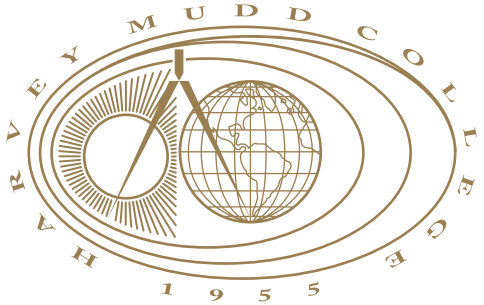
2001

Computational Evolutionary Linguistics

Tracy vanCort
Harvey Mudd College



Computational Evolutionary Linguistics

Tree-Based Models of Language Change

by

A.T. van Cort

Professor Elizabeth Sweedyk, HMC Computer Science, Advisor

Professor Karen Kossuth, Pomona Linguistics, Advisor

Advisor: _____

Advisor: _____

Reader: _____

May 2001

Department of Mathematics

HARVEY MUDD COLLEGE

Abstract

Computational Evolutionary Linguistics

Tree-Based Models of Language Change

by A.T. van Cort

May 2001

Languages and species both evolve by a process of repeated divergences, which can be described with the branching of a phylogenetic tree or phylogeny. Taking advantage of this fact, it is possible to study language change using computational tree-building techniques developed for evolutionary biology. Mathematical approaches to the construction of phylogenies fall into two major categories: character-based and distance-based methods. Character-based methods were used in prior work in the application of phylogenetic methods to the Indo-European family of languages by researchers at the University of Pennsylvania. Discussion of the limitations of character-based models leads to a similar presentation of distance-based models. We present an adaptation of these methods to linguistic data, and the phylogenies generated by applying these methods to several modern Germanic languages and Spanish. We conclude that distance-based for phylogenies are useful for historical linguistic reconstruction, and that it would be useful to extend existing tree-drawing methods to better model the evolutionary effects of language contact.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Every thesis is special to its author, but this one perhaps more so than usual. It provided me with an opportunity to do interdisciplinary research on a topic of great personal interest while surrounded by incredible resources, including some of the most amazing people I could have ever hoped to exchange ideas with. That said, I am very grateful to the Harvey Mudd College Mathematics and Pomona College Linguistics Departments for allowing me to write a dual thesis, and my advisors, Professors Z, Kossuth, and Levin, for their advice and encouragement. I would also like to thank Professor Carmen Fought of Pitzer College for advising me to pursue this idea when I was supposed to be doing research with her, and providing second opinions on some tough phonology problems. Professor Lesley Ward and the student and faculty members of the Math 197 class offered valuable suggestions and insightful questions. Finally, special thanks are due to Peter Boothe, for more reasons than I can list, but especially friendship and love.

This thesis is dedicated to the members of all my family trees.

Chapter 1

Introduction

Phylogenetic trees, or phylogenies, represent the relationships between species in evolutionary biology. Various mathematical methods can be used to construct trees for groups of species that are known or believed to be related. Phylogenies generated in this way can be used to draw conclusions about the evolutionary histories of the species being investigated. Language families have historically been described with trees as well, but the methods used to build these are considerably less formalized. The methods used to construct evolutionary trees for species could be a valuable tool for addressing problems in historical linguistics. The background provided in this chapter is intended to introduce basic concepts and motivate further discussion of languages as species and tree-based descriptions of language change, outlined in Section 1.4.

1.1 Languages and Species

Does it make sense to approach languages with the computational machinery developed for species? Aside from the fact that it would be useful to do so, languages and species have many similarities. Both are difficult to define, as they manifest themselves at the population level, where classification is often arbitrary, difficult, or ambiguous. For example, artificial selection by humans has amplified and modified naturally occurring variation among members of species to produce strikingly different breeds of domesticated animals. Consider dogs: Chihuahuas and Great Danes are still classified as members of the same species. Similarly, many dialects of English are so phonetically and syntactically different as to be mutually unintelligible at first; yet these all share a common writing system. Linguists often use the saying "a language is a dialect with an army and a navy" to express the fact that social and political distinctions often play a role in determining boundaries between languages; in ambiguous cases, biological species may be differentiated subjectively. Early systems of classification distinguished species primarily by their morphological characteristics; Charles Darwin first suggested an evolutionary interpreta-

tion of taxonomic hierarchies (and in fact pioneered the use of trees for their representation). Still more modern species definitions focus on genetic relationships between the populations in question. As a result, there are still controversies as to exactly where species boundaries should be drawn. Among African equines, the quagga (*Equus* or *Hippotigris quagga*), which is striped only on its head, neck, and shoulders, is sometimes classified as a subspecies of zebra (*E. burchelli*) and sometimes as something else entirely (hence the ambiguous binomial nomenclature). For more information about distinctions between languages, see Chapter 25 of [33]; a good historical discussion of the biological definition of species can be found in Chapter 2 of [39].

Species and languages have certain attractive structural similarities as well: species are populations made up of subpopulations made up of individuals, and languages are dialects made up of subdialects made up of ideolects spoken by individuals. Not all individuals of a population are identical, just as there is variation among species and languages. In both languages and species, variation may be random, geographically distributed, or determined by some kind of outside pressure. For example, English teachers and other language mavens may enforce certain standards of grammar, usage, and pronunciations, just as dog breeders enforce certain traits by artificially selecting for them. Variation among individuals and groups is a source of evolutionary change for both species and languages. Finally, when species are defined as reproductively isolated populations—that is, groups that cannot produce viable offspring by interbreeding—languages can be analogously defined as mutually unintelligible dialects. Then, just as different subpopulations of a species might diverge into reproductively isolated groups, the subdialects of a language might become mutually unintelligible as well. The branching of a tree is a useful way of representing a sequence of evolutionary divergences. Historically, linguists and biologists have both used trees to describe evolution; however the methods by which those trees were arrived at, evaluated, and interpreted have been quite different.

1.2 Trees in Historical Linguistics

In historical linguistics, trees represent the results of reconstruction: efforts to discover the origins and evolution of modern languages by deducing the features of their ancestors, called **protolanguages**. Evidence from historical writings sometimes contributes to what is known about ancient predecessors to modern dialects, just as morphological data from the fossil record is of-

and phonology are represented as siblings on a tree, descended from a protolanguage with features like those the two descendant languages have in common. Relationships between more dissimilar languages are harder to deduce, but eventually similarities between reconstructed protolanguages, proto-protolanguages, and so on serve to connect all the languages of the world into about twelve language families without too much difficulty. Further connections are more controversial, but researchers such as Merrit Ruhlen of Stanford seek to prove through reconstruction that all of the world's languages share a common origin (this is referred to as the theory of **monogenesis**) [43]. The Comparative Method of historical linguistics, published by Hoenigswald in 1960, formalizes the process of reconstruction with specific rules for developing correspondences between sets of features of known languages and the hypothesized feature sets of ancestral languages [32].

1.3 Phylogenies in Evolutionary Biology

A group of biological species descended from a common ancestor can be represented as the leaves of a tree whose root is the shared ancestor, much as a family tree might represent the relationships between family members descended from a common ancestor. If nothing is known about the order in which they diverged, the descendant species are drawn as a star phylogeny, such as Figure 1.1. More informative trees, such as Figure 1.2, reflect the order in which species diverged and contain intermediate nodes representing ancestral species between ancestor and modern-day descendants.

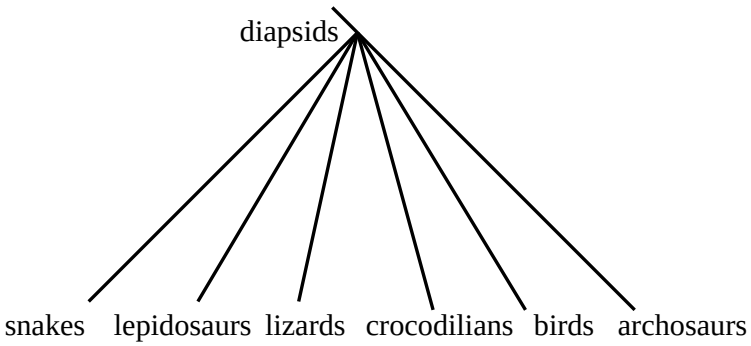


Figure 1.1: Star phylogeny for modern birds and reptiles and their ancestors.

The vocabulary used to describe phylogenetic trees is straightforward: the ancestral node is

decended from a common ancestor is called a **monophyletic group** or **clade**. In this paper, I will occasionally refer to clades with respect to particular ancestral nodes for clarity. For example, in Figure 1.2, snakes and lizards are a clade with respect to lepidosaurs. In contrast, reptiles are a **paraphyletic group**. Though descended from a common ancestor, in a more immediate sense the snakes and lizards are members of a clade with respect to lepidosaurs, whereas alligators are more closely related to modern birds, with whom they share membership in a clade with respect to archosaurs. Biologists use the term **gens** for an evolutionary lineage; this could be represented as a sequence of vertices descending from the root of a phylogeny. Finally, biologists generally show phylogenies "growing" from the root up, although they also use top-rooted trees or trees with the root to the left and leaves to the right, whereas linguists almost always adhere to the top-rooted tree convention.

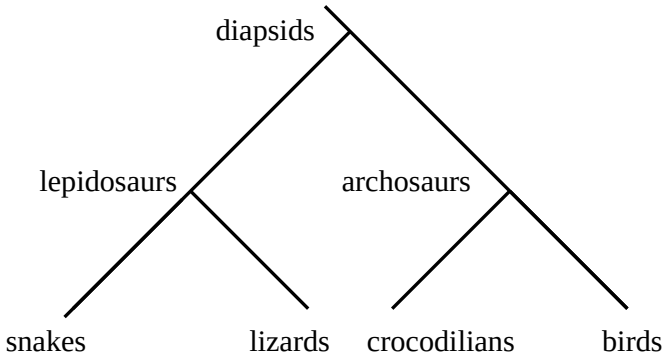


Figure 1.2: A more informative phylogeny for the species in Figure 1.1

1.3.1 Tree Construction and Evaluation

There are two major classes of mathematical models for tree construction in evolutionary biology: the **character-based methods** and the **distance-based methods**. Character-based methods describe species in terms of their features, and construct phylogenies by comparing characters across species, and evaluate the goodness of the resultant trees in terms of the behavior of those characters. Distance-based methods stem from the idea of a distance measure between two species, and try to construct trees whose branch distances most closely match the observed distances determined from data taken about the species in question. In both methods, it is trivial to describe the species being studied as a matrix of features or distances given a tree describing

NP-Complete or NP-Hard no matter what class of method is applied [38]. Thus it is necessary to apply heuristics and approximation methods to solve these problems in a reasonable time frame.

1.4 Thesis Contents

Constructing a phylogeny can be a way of inferrring the relationships between a group of species that are believed to be related, but whose evolutionary history is unknown. We can take advantage of the similarities between languages and species to apply mathematical methods developed for the construction of phylogenetic trees to the problem of describing language change. researched the possibility of using character and distance-based methods to construct phylogenies for languages. Chapter 2 discusses character-based methods in general and with respect to the classical methods of historical linguistics, presents examples of commonly used character-based methods and the results of prior work in the linguistic application of character-based tree-building techniques. The Computational Historical Linguistics Project (computer scientist Tandy Warnow and linguists Don Ringe and Ann Taylor at the University of Pennsylvania) used the character-based method of perfect phylogeny in constructing their evolutionary tree for Indo-European and its descendants [48]. I then present the limitations of character-based models, especially for modeling languages, and introduce the notion of distance-based trees by comparison to character-based methods in Chapter 3. In Chapter 4 I describe distance-based tree building methods at greater length and introduce my work in adapting those methods to languages for the purpose of reconstructing a phylogeny of several modern Germanic languages and Spanish. Chapter 5 presents and analyzes my results. I conclude in Chapter 6 that distance-based trees are a valid method of determining evolutionary trees for languages and that vowels are a valid source of historical data, and suggest areas of possible further research.

Chapter 2

Character-Based Trees

Character-based techniques for building phylogenetic trees model species as sets of features. Features common to all the species being compared are called **characters**, and the species-specific manifestation of these characters are called **character states**. For example, if the character being compared is forelimbs, character states might include human arms and hands, dolphins' fins, horses' hooves, and bats' wings. Linguistic characters could include the basic word order of a sentence in a language, with character states subject-object-verb, subject-verb-object, and so on. Character-based methods construct phyloegenies for the evolutionary history of a group of species by comparing the character states of the group's members. These methods are intuitively appealing and similar to the traditional methods of historical linguistics described in Section 2.1, so they lend themselves easily to linguistic applications. The mathematical character-based methods of parsimony and compatibility are described and discussed in Sections 2.2.1 and 2.2.2. Compatibility, or perfect phylogeny, features prominently in prior work in the linguistic application of mathematical tree-drawing techniques: it was used by the Computational Historical Linguistics Project in constructing their evolutionary tree for Indo-European and its descendants [48]. Their work is summarized in Section 2.3.

2.1 Linguistic Methods

The intuitive appeal of character-based methods is that species are easy to describe in terms of their features. Any five-year-old can tell you a dog is a furry animal with four legs, a tail, and a proclivity for making noises like "woof". Linguistic features are not as readily apparent, but all languages can be described in terms of their vocabulary and grammatical rules (lexical and syntactic characters, respectively). Other important features of languages include the set of sounds in their phonologies, or handshapes in the sign language equivalent, as well as the linguistic environments in which those sounds or handshapes occur.

respond across known languages: these are just like characters, which are analogous across species. By comparing these across **cognates**, words descended from the same roots, linguists develop hypotheses about the phonetic and phonological changes or semantic shifts that caused the protolanguage to diverge into its descendants, and reverse these to determine the features of the ancestral language. On a tree, descendants are shown as branching off from the node for the protolanguage. **Loanwords**, vocabulary items introduced from other languages, can not be compared in this way, as they entered the lexicon as a result of language contact, a process quite different from sound or semantic shifting. Another reconstruction technique commonly used by historical linguists is **subgrouping by shared innovations**. This involves taking a feature inventory of the languages being studied and grouping together languages that share a statistically significant number of features (in particular features which differentiate them from the other languages in the set and are unlikely to have arisen or disappear randomly; these are called innovations).

For an example of a simple reconstruction and the tree which corresponds to it, consider the languages A, B, and C, compared on ten features as shown in Table 2.1.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
A	a	d	f	f	j	j	o	r	t	t
B	b	e	g	g	k	m	p	p	u	w
C	c	c	h	i	l	n	q	s	v	v

Table 2.1: Sample data for the reconstruction shown in Figures 2.1 and 2.2

The protolanguage all three are descended from must contain all ten features being compared in some form. Using the method of grouping through shared characteristics, first note that in Language A, features (3) and (4), (5) and (6), and (9) and (10) are indistinguishable, as (3) and (4) and (7) and (8) are in Language B, and (1) and (2) and (9) and (10) in Language C. Pairs which are indistinguishable in only one language—(1) and (2), (5) and (6), and (7) and (8)—do not contribute to the reconstruction. However, since (3) and (4) are indistinguishable in Languages A and B, those two are likely to form a subgrouping of the three languages, for the tree shown in Figure 2.1. Since (9) and (10) are indistinguishable in Languages B and C, another possibility is for the latter pair to be a subgrouping, as shown in Figure 2.1. (Example taken from [32].)

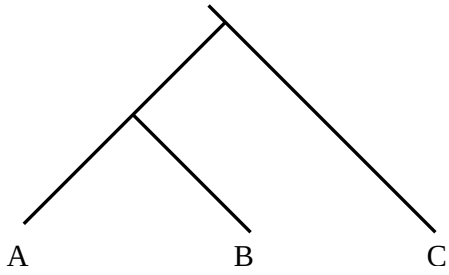


Figure 2.1: One possible reconstruction of the data in Table 2.1

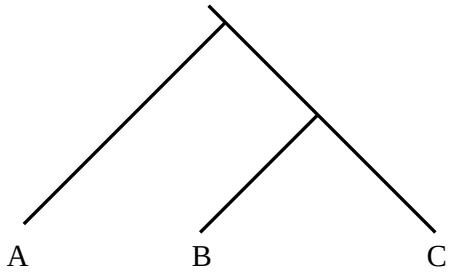


Figure 2.2: Another possible reconstruction of the data in Table 2.1

arrive at multiple trees for one data set. Figures 2.1 and 2.1 are an example of **overlap**, in which one species shares an equal number of innovations with multiple species, such that it is impossible to determine the sequence of changes. Overlaps can sometimes be resolved by comparing more characters to clarify groupings by discovering additional correspondences.

2.2 Computational Methods

Techniques for determining evolutionary trees by examining the features of a group of biological species are considerably formalized than linguistic methods. The generalized formulation of a character-based method is as follows:

Definition 1 *First, define a* **character set** $\mathcal{C} = \{1,\ldots, m\}$ *of features on which a set of* n *species is compared. Each species* s *is represented as a vector* $(s_1,\ldots, s_m)$ *where* s_c *represents the state of the* c*th character. Then* $s_c \in \mathcal{A}_c = \{1,\ldots, r_c\}$*, the set of possible character states for a character* c*. Given a set* $\mathcal{S}$ *of* n *distinct species of* m *characters, the objective is to construct a phylogenetic tree* T *describing the relationships among those* n *species according to some criterion.*

Usually some function on the tree which must be maximized or minimized, these criteria reflect

sider the assumptions of the frequently used character methods of parsimony and compatibility, described in Sections 2.2.1 and 2.2.2, respectively.

2.2.1 Parsimony

The basic assumption of parsimony-based tree-drawing methods (also called minimal evolution methods) first developed by Camin and Sokal in 1965, is that evolutionary changes, modeled as changes in the character state of a particular character, are very rare [5]. So the goal of these methods is to construct phylogenies with the fewest possible character state changes. In particular, the principle of maximum parsimony holds that evolutionary **reversals** are extremely unlikely. So when a descendant species diverges from its ancestor on a character, it is highly unlikely for that character to revert back to the ancestral state. To return to the introduction's example of modern birds and reptiles and their ancestors, lizards and crocodilians have four legs, whereas birds have two legs and wings, and snakes have none. The most parsimonious tree for this character would be one in which all four modern species were descended from a four-legged ancestor, and there were a total of two evolutionary changes: one in which the snakes lost their legs, and another in which birds' forelimbs evolved into wings, illustrated in Figure 2.3.

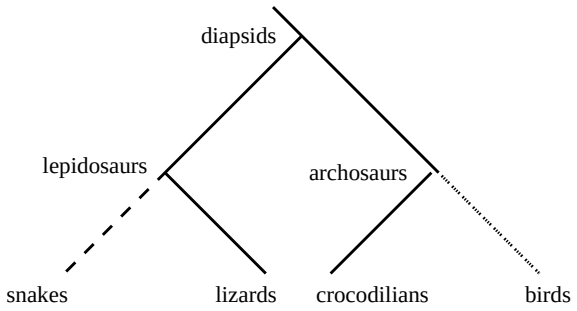


Figure 2.3: A parsimonious tree for modern reptiles and birds on the character of leg number. Contains two changes and no reversals.

A less parsimonious scenario for the evolution of these species on the same tree structure would be if diapsids or one of their ancestors had two legs and wings, which then evolved into four legs for archosaurs, lepidosaurs, and most of the modern reptiles, and into no legs for snakes, and back into two legs and wings for birds (Figure 2.4).

To avoid exhaustively searching all possible trees, maximum parsimony-based sofware pack-

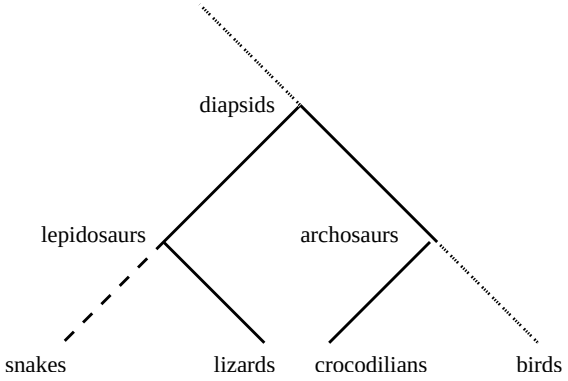


Figure 2.4: A less parsimonious phylogeny for modern reptiles and birds. Contains three or four changes (depending on the character state of the diapsids) and one reversal.

approach curtails the number of trees constructed and evaluated, it is nonetheless prohibitively slow for comparisons of more than 10 species [30]. In fact, the computational complexity of maximum-parsimony methods was first discussed in 1982 [26] and determined to be NP-Complete in 1983 [11]. Furthermore, like all character-based methods (and the traditional historical linguistics techniques), it may produce several very different-looking but equally parsimonious trees for the same data. Branch and bound may also rule out valid trees because of early violations of parsimony, so it may be necessary to vary the order of its input to find all possible maximum-parsimony trees, and running the algorithm repeatedly may defeat the purpose of having a faster algorithm. Other non-exponential methods of determining parsimonious phylogenies are based on heuristics, and the order of the input must be varied in these as well.

2.2.2 Compatibility

Compatibility, or perfect phylogeny, is a special case of parsimony based on the assumption of character state changes so rare that it is extremely unlikely for the exact same character state to evolve independently in different species. Under this model of evolution, new character states generally arise only once and are passed on to descendant species [23].

Definition 1 *An evolutionary tree T on a set of n species $\mathcal{S}$ is called a* **perfect phylogeny** *for $\mathcal{S}$ if it has contains a vertex for every member of the species set S (and in particular all leaves of the tree are elements of S), and all species containing a certain character state c_j for a character c induce a subtree of T. If all*

Note that any reversals in a perfect phylogeny T would create disjoint subtrees, hence T contains no reversals and as such is highly parsimonious. The perfect phylogeny problem: determining whether a set of species $\mathcal{S}$ has a perfect phylogeny T was shown to be NP-Complete by Bodlaender et al. [28] and independently by Steel [45] but polynomial-time solutions have been found by restricting the number of characters [3, 34, 22] and character states [1, 27].

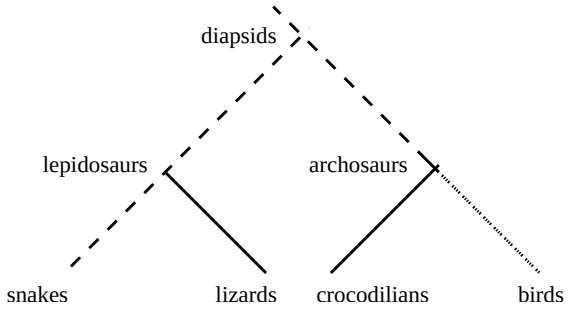


Figure 2.5: A reversal-free tree for the modern reptiles and birds that is not a perfect phylogeny. Contains three evolutionary changes and no reversals.

Of the sample trees given for parsimony, note that Figure 2.3 is a perfect phylogeny, as all four-legged species form a connected subtree, and birds and snakes are each on their own subtrees (albeit leaves). Figure 2.4 is not a perfect phylogeny, as it contains a reversal. Likewise, if diapsids, archosaurs, and lepidosaurs were all legless, like snakes (Figure 2.5), the same tree structure common to Figures 2.3, 2.4, and 2.5 would not generate a perfect phylogeny, as the four-legged lizards and crocodilians would be disjoint. Note that this tree is also not as parsimonious as that of Figure 2.4.

2.3 The Computational Historical Linguistics Project

The Computational Historical Linguistics Project (CHLP) presented a character-based evolutionary tree for the Indo-European language family at the National Academy of Science's November 1995 Frontiers of Science Symposium [48]. Twelve Indo-European language families were modeled with data from the most well-studied member of each, and tested on lexical and phonological characters based on a vocabulary list established in [47]. The researchers found several perfect phylogenies for these languages using a program based on [1], but they had to remove the German subtree from the data to do so. The Germanic tree was inserted later and tested in

the tree shown in Figure 2.5.

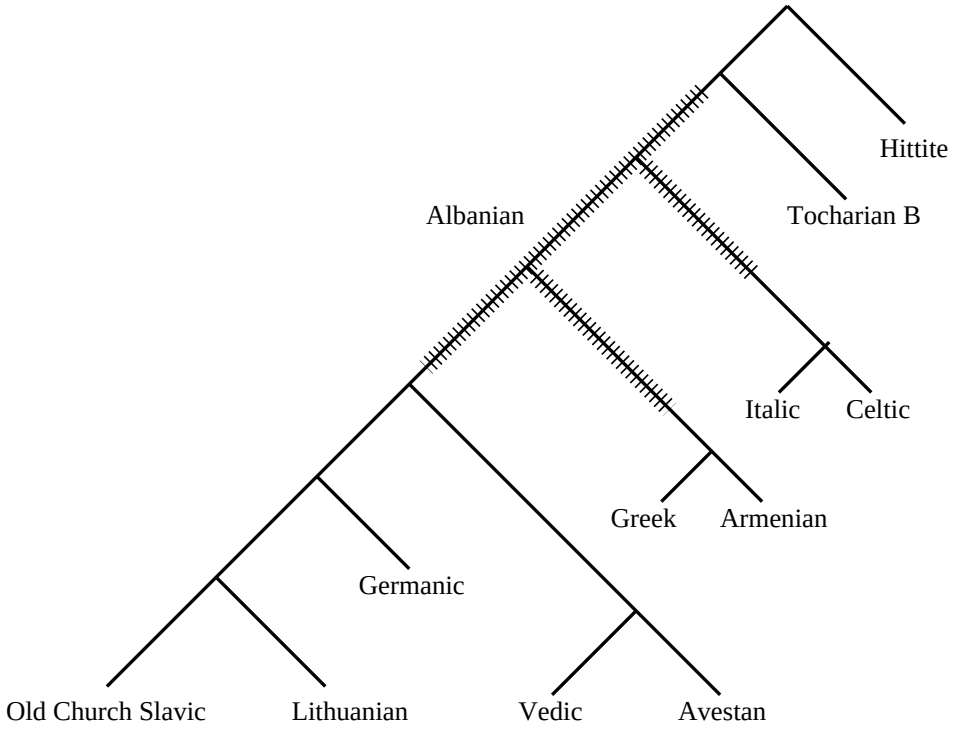


Figure 2.6: The CHLP's phylogeny-based evolutionary tree for Indo-European languages. Not to scale, no time except nodes higher on the tree represent earlier branchings. A leaf for Albanian may be located along any of the shaded edges.

Besides the difficulties accounting for the Germanic languages and Albanian, which appeared in several positions on equally parsimonious trees, Warnow et al. had to contend with the phenomenon of **polymorphism**, multiple character states for a single character in one language. The English words "big" and "large" are an example: they differ only slightly and subtly in their meaning, and many languages do not distinguish between the two. An example of phonetic polymorphism would be free variation between two pronunciations of a word (for example, in many dialects of English, **[æn]** is used interchangeably with **[ænd]** for "and"). Polymorphism poses a problem to character-based methods since these models can only account for one character state per character. These problems were eventually resolved by using techniques developed by Warnow and other computer scientists to build consensus trees between the phylogenies that differed due to polymorphic characters [37].

Responses to the CHLP's results were mixed, as the early evolutionary history of Indo-

Two such controversies addressed by Warnow et al. Indo-Hittite hypothesis, which states that Anatolian (represented by Hittite) is the first family to branch from the main Indo-European root, and the Italo-Celtic hypothesis, which claims that those two should be sisters, with no other siblings. The CHLP's results supported both of these, which met with much opposition. They were also criticized for the ambiguity of their Germanic and Albanian results, which some considered tantamount to invalidating the rest of the phylogeny.

2.4 Limitations of Character-Based Models

Character-based models, though intuitively appealing, may not be the best method of describing languages and generating phylogenetic trees that accurately describe their interrelationships and evolutionary histories. For one, parsimony and compatibility have been shown to produce incorrect results if the species being studied evolve at different rates, as languages often do [17]. Second, there is no guarantee that they ever converge to one best tree for the criterion being evaluated. Finally, character-based models provide no means of describing the effects of **language contact**, the ways in which a language may change when its speakers come into contact with people who speak another languages. (The last two problems are also among the drawbacks of the historical methods.) In the following chapter, I compare character-based methods to distance-based tree-building techniques, an alternative approach that provides solutions to some of the problems with character-based trees, and was the basis for my original work in this thesis.

Chapter 3

Character or Distance Methods?

As mentioned briefly at the end of the last chapter, the limitations of traditional techniques of historical linguistics and character-based models for building phylogenetic trees may be extremely problematic in the study of language change. Fortunately, there are other approaches to the problem of constructing accurate phylogenies for species. I studied distance-based methods of tree construction, discussed in greater detail in Chapter 4, as an alternative to character methods. For the purposes of this comparison, it suffices to state that these methods describe a group of species with a matrix of pairwise distances between them, as shown in Figure 3.1. The (i,j)th entry of this **distance matrix** is simply the distance between the ith and $j$the species being compared. The specific definition of distance may vary, but a few different measures are discussed in Chapter 4, but for now note that the distances are symmetric lower bounds, and a species is always zero distance from itself. The methods then attempt to build trees whose branch lengths or weights most closely match these distances, with attention to the fact that the data used to determine distances (or even the distance metric itself) may not be reliable.

This chapter presents several problems that pose a challenge to character methods, and ways in which these problems can be addressed with distance methods, if any exist.

$$\begin{bmatrix} 0 & d_{12} & \dots & d_{in} \\ d_{21} & 0 & & d_{2n} \\ \vdots & & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & 0 \end{bmatrix}$$

Figure 3.1: Generic distance matrix for n species.

3.1 Rates of Change

As mentioned in the conclusion of the last chapter, character methods have been shown to give highly inaccurate results when the species being compared evolve at different rates. Felsenstein first analyzed this problem in [17] by modeling evolution as a stochastic process on characters and assigning a different probability of change to the characters of each species, then simulated their evolution over time. For the trees he generated, the probability that parsimony and compatibility methods returned an incorrect phylogeny increased to nearly 100% as a polynomial function of the difference in the species' rates of change. Felsenstein's results were generalized by Hendy and Penny in [31].

This is a problem in the study of language change because languages, like biological populations, often evolve at unequal rates. For example, South Africa was settled by Dutch colonists during the 17th century. The settlers' language evolved into Afrikaans, which is often unintelligible to speakers of modern Dutch, and seems far more removed from the linguistic ancestor both modern languages descended from. On the other hand, Icelandic has evolved so slowly that Iceland's constitution, which predates the origin of Afrikaans by several centuries, is still in use today. All three of these languages are part of the Germanic family that presented a problem to Warnow et al.'s phylogeny, and which I chose to study with distance methods.

Distance-based methods are an improvement over character methods for dealing with these problems because the measures used to determine distances are likely to reflect differential rates of change. Granted, character data would be more likely to show more changes for a rapidly-evolving species, but it is possible for distance methods to respond to these distances reflecting differential rates of change, by simply assigning longer branches to more distant species. Similarly, unlike character methods, distance methods do not assume minimal evolution, and so are not as highly sensitive to generating trees with numerous character state changes or even reversals, which are more likely to occur in rapidly-evolving species. Although any distance method can be modified to include the assumption of a built-in "evolutionary clock" that enforces a constant rate of character state change on all characters of all species, the default for these methods is to assume differing rates of change and treat distances between species as evolving accord-

3.2 Consistency

Another serious problem with character-based methods is the likelihood of their producing inconsistent results: several different, yet equally parsimonious or compatible phylogenies might be generated from the same data set [17, 31]. Although consensus tree methods can be used to combine the features of several different character-based trees into one, this only adds to the computational complexity of the problem and can lead to ambiguous results like the ones for which the CHLP researchers were criticized [37]. A simple solution to the consistency problem would be extremely valuable in historical linguistics, where problems like overlap create inconsistencies in the results of traditional reconstructive methods.

Distance-based methods provide such a solution: they have been shown to converge to a single best tree given sufficient data [12]. Farach and Kannan modeled evolution as a simple stochastic process and developed a measure for distances between evolutionary trees. They then used a simple distance-based method to essentially reverse the evolutionary process, and showed that their algorithm was guaranteed to produce a phylogeny that could be brought arbitrarily close to a tree representing the steps of the evolutionary process, using a method of tree comparison developed in an earlier paper of Farach and Mikkel Thorup [13]. The Farach-Kannan proof, published along with a result suggesting character-based data for a group of species can be converted into distance matrices without significant loss of information, strongly supports the use of distance methods (as always, assuming the researchers' model of evolution was a reasonable one). Character methods have never been shown to converge.

3.3 Language Contact

The final problem common to both traditional linguistic techniques and character and distance-based methods is that of describing the effects of language contact, a known source of linguistic variation and evolution. Strictly genetic models of language change often fail to account for language contact, just as biological approaches to phylogeny often choose to ignore the possibility of horizontal gene transfer, such as exchange of genetic material between members of the same generation or different species of bacteria through plasmid exchange. But whereas horizontal gene transfer is relatively rare in biology, contact between languages is more frequent, and is an important source of language change. Unfortunately, it is particularly problematic to traditional

ous example, one reason Afrikaans may be more different from 17th-century Dutch than modern Dutch is that the Boers came into contact with numerous native South African languages, as well as the language of English settlers. The Icelandic constitution, on the other hand, was written in A.D. 930 in a language whose speakers were relatively isolated for centuries. Isolated populations have fewer opportunities to borrow words from other languages into their speech.

Both character and distance models can begin to address the problem of describing language contact by incorporating data about likely loanwords, which could be weighted to reflect its varying importance in different situations. In extremely isolated cases like that of Icelandic, the effects of language contact could probably be approximated as zero, so even traditional linguistic methods could be applied. However, contact plays some role in the development of most languages, and might be critical to understanding language change in situations involving frequent immigration, colonization, conquest, or other historical events likely to create interactions between speakers of multiple languages. The most extreme cases of language contact present the greatest challenge to all tree-based studies of historical linguistics, regardless of method: the problem of **contact languages** like **pidgins** and **creoles**. The former are codes created out of necessity by speakers of multiple languages with none in common; the latter are the languages that grow out of such codes if children grow up speaking it [2]. Although the former are likely to be very simple and impoverished in its features, the latter are full-fledged languages in their own right—but they are unlikely to appear on any tree. The model of evolution assumed by all phylogenetic methods is that of repeated divergences, which the formation of contact languages violates by definition. No model of language evolution would be complete without some discussion of language contact; however, as Warnow et al. noted, pidginization and creolization are fairly rare, and the CHLP's character-based study seems to have been fairly successful despite their having overlooked the possibility of contact's effect on their results. For more general information on language contact, pidgins, and creoles, see [2]; a good discussion of the problem of language contact in historical linguistics can be found in [46].

3.4 Distance-Based Trees for Language Change?

Judging by this cursory analysis, distance-based methods offer several advantages over character-based models for the study of language change. First and most importantly, they have been

lems with the results of character-based methods. Second, distance methods offer greater flexibility in describing situations of differing evolutionary rates, which are not unusual in historical linguistics due to various social and political factors. Next, they are certainly no worse than character-based methods at addressing the problem of language contact. I believe both classes of methods are capable of addressing weak cases of language contact, like borrowing of vocabulary items across languages, which in turn may lead to a greater understanding of the stronger cases, like contact languages, and possible extensions of the model. Finally, as far as I know distance-based methods have never been applied to languages, and so there is almost certainly something to be learned by testing the use of these methods for the study of language change.

Chapter 4

Distance-Based Trees

Another major class of approaches to reconstructing phylogenies are the distance-based methods. As mentioned briefly in the previous chapter, which presented the reasons I decided to try adapting distance-based trees as an alternative to character-based methods for the study of language change, these methods represent a group of species as a matrix of pairwise distances between them. Such a matrix is trivial to determine if the evolutionary tree for the species is known, simply by assigning branch lengths. The problem of determining a tree given a matrix of distances is NP-hard in general, and NP-Complete in most cases [38]. Part of the problem of determining a distance-based tree is the fact that most distances are an approximation of the relationships between the species in question (usually assumed to be a lower bound). So determining a distance measure has been extensively studied as well. It has been shown that distances can be generated from genetic chararacter data without losing too much information about the species [12]. This chapter presents a bit more general backgound on these methods and the methods by which I adapted them to develop a phylogeny for several modern Germanic languages and Spanish.

4.1 Basic Problem and Ideas

Distance-based methods work by defining a distance metric on the set of species being analyzed, and constructing a distance matrix according to the differences between pairs of species. The distances in the matrix are generally considered lower bounds. They are based on a sample of the populations being compared, and a final criterion is that they converge to the total distance between the species as sample size increases to total species size. The tree or trees that best preserves those distances is considered the best model of the evolutionary history of the species in question. The problem of finding such a tree can be formulated in many different ways, but most of them are NP-Complete or NP-Hard [38]. Distance-based trees have been extensively

have been proposed. Furthermore, many of these methods have been implemented, so I took advantage of PHYLIP, an existing and extensively documented tree-construction software package [20] and attempted to build a distance-based phylogeny of my own. I chose to study several modern Germanic languages: English, Dutch, German, Icelandic, and Norwegian. I chose these languages (and Spanish, a late addition which served as an outgroup to contrast with the others) because I speak two of them, and Warnow et al [48] ran into problems trying to fit the Germanic subtree in with the rest of their phylogeny for Indo-European, so I was curious to see if this subtree behaved in any unusual way. But I digress. On to the methods.

4.2 Heuristics and Approximation Methods

The following are descriptions of the distance methods I used in constructing my trees. It is important to note that there are many others, and that different methods may produce different results from the same data. In fact, just as with character-based methods, some heuristics will produce different trees from the same data entered in a different order, just as the path by which one escapes from a maze varies depending on what rule of thumb is used. No one method is recognized as the best, so I chose to compare results for two well-known and frequently used methods I felt I understood fairly well.

4.2.1 Least Squares Methods

Like the generic character-based method described in Chapter 2, one way of constructing distance-based phylogenies is to evaluate a function on likely trees for the given data. The least-squares family of methods, first introduced by Fitch and Margoliash in 1967 [21], all involve trying to minimize a sum of squares function of the following form, where $D_{ij} =$ is the observed distance between species i and j (as found in the distance matrix) and $d_{ij} =$ expected distance between species i and j (as found in the tree).

$$\sum_i \sum_j \frac{(D_{ij} - d_{ij})^2}{D_{ij}^p}$$

Different researchers have endorsed different values of p. For example, Cavalli-Sforza and Edwards [7] set p = 0 in cases of low measurement error, reducing the denominator to 1 and relying solely on the difference of squares to gauge a tree's accuracy. I used the method of Fitch

[21]. Variations on this method sometimes assume a "molecular clock"—that is, all species being compared are each other's contemporaries, and evolved at the same rate. The distances involved in methods of this sort are called **ultrametric**.

4.2.2 Neighbor-Joining

A newer tree construction method is a greedy heuristic called neighbor-joining or nearest-neighbor. Originally developed by Nei and Saitou [44], this method has been steadily increasing in popularity because of its speed. Unlike many other distance matrix and character-based methods, neighbor-joining does not involve an exhaustive search. The basic steps of the algorithm are as follows:

1. Search the distance matrix for the smallest nonzero between a pair of species (every species is zero distance from itself). These two will be each other's nearest neighbors in the tree, so join them at a node.

2. Replace the neighbors' entries in the distance matrix with an entry for the node connecting them. Find distance values from that node to the other species by averaging the neighbors' distances.

3. Repeat Steps 1 and 2 on the new distance matrix. Continue until only one node (the ancestral node for all the species) is left.

See Chapter 2 of [6] for a step-by-step example of a tree "grown" in this manner.

4.3 Defining a Metric on Languages

Before any of these methods could be applied, however, I needed to construct distances between the languages being studied. In evolution, distances between species are usually defined in terms of genetics. One very simple measure of genetic distance is the percentage of genes shared by two species; slightly more sophisticated models address actual gene structure a bit more precisely. In general, the genome of a species is made up of strings of bases, which, if they obey certain rules, encode amino acids, which can be strung together according to certain rules to form proteins, which in turn may obey certain rules to form tissues and so on up to the organismal,

interpreted as words if they obey certain rules, and in turn there are rules for stringing together words to encode meaning. So I decided to define distance between languages in terms of the phonemes underlying their words and rules.

Definition 1 *A* **sound change** *is any alteration in the phonetic features of a phoneme causing it to be recognized and interpreted as another, or none at all.*

Definition 2 *The* **phonetic distance** *between two languages is the average number of sound changes required to transform a word in one language into that word's equivalent in the other.*

I hypothesized that the more closely related two languages are, the more similar they sound. Just as biologists compare gene sequences for the same function, I compared words with the same meanings. The meanings in question were a set of basic vocabulary items (see Appendix A). At first I thought to create a vocabulary set based on words frequently used in the languages I was studying, but word frequency lists are notoriously unreliable (one for English had "bad" in the top 40 words twice) and differ across languages anyway. So I built a list of basic vocabulary from the ground up. Ideally such a list should reflect the structure of the languages being studied (percentages of various parts of speech, percentages of words borrowed from other languages, etc.) My list concentrates primarily on nouns, but I did make an effort to take loanwords into account in computing distances.

4.4 Vowel Distance

Definition 3 *The* **vowel distance** *between two languages is the average phonemic distance between the vowels of a word in one language and its equivalent in the other.*

I chose to measure distance in phonemic terms because of the analogy to genetics and because the sounds of a language are relatively easy to quantify. In particular, vowels lend themselves easily to a relatively simple encoding in terms of their phonetic features (see Table 4.4, based in part on the system described in [9]). The phonetic features varied in vowel sounds include the position of the tongue at their articulation (front to back, high to low), as well as in length and rounding (the shape of the lips when spoken). Vowels also dominate syllables and color the overall sound of a language. Although traditionally historical reconstruction has focused on the consonants, on the premise that vowels are too subject to rapid change to be informative, biol-

species, so I decided to use vowels from speech samples to determine a "phonemic snapshot" of the languages. As for the effect of dialect or accent differences, it is important to keep in mind that my trees reconstruct the evolution of the particular ideolects I sampled, which is likely but not guaranteed to mimic the evolution of the overall languages in question. Biologists run the same risk in using data sampled from individual organisms to represent a population.

	Front	**Central**	**Back**
high	i (d,e,g,i,n,s); y (d,g,i,n)		u (d,e,g,i,n,s)
	ɪ (d,e,g,i); ʏ (g,i)		ʊ (e,g)
mid	e (d,e,g,i,n,s); ø (d,g)		o (d,e,g,i,n,s)
	ɛ (d,e,g,i,n); œ (d,g,n)	ə (d,e,g,i,n)	ɔ (d,e,g,i,n,s)
low	æ (e)		
	a (d,g,s)	ʌ (e,i)	ɑ (d)

Table 4.1: Vowels in English, Dutch, German, Icelandic, Norwegian, and Spanish.

Notes on Table 4.1: front vowels are unrounded by default; back vowels are rounded. In pairs, the vowel to the left is unrounded. Lowercase letters in parentheses indicate the languages in which each sound appears.

4.5 Method

I developed a feature-based encoding of all the vowels in the languages I was studying, the idea being to approximate phonemic distance with vowel distance through feature by feature comparisons of the vowels of word pairs with the same meaning. My encoding was based on the vowel chart in Table 4.4 and the numerical encoding was incorporated into a feature dictionary used in my Python programs for computing vowel distances between word sets (see Appendix C for source code). Basic feature vectors for all consonants appearing in the vocabulary sets was encoded in a separate dictionary, and the distance finding programs tested to make sure all sounds in the vocabulary set were in one of those two dictionaries, to avoid accidentally overlooking vowels that had not been encoded (see Appendix D for vowel and consonant feature

4.6 Metrics and Special Cases

Not all word pairs can simply be compared vowel for vowel. Some have different numbers of vowels, either because syllables have been lost or gained over time, or because a loanword with a different number of syllables replaced ancestral lexical items. Finally, single vowels were easy to encode, but what about diphthongs, sounds produced by gliding an initial vowel into another? The different metrics implemented in the distance finder addressed these problems, as well as the fact that longer words are more likely to be a greater absolute distance apart. Each of these problems and my approaches to solving them (generally by varying the method of generating distances) is discussed in the following subsections.

4.6.1 Vowel Number

Longer words are likely to generate greater vowel distances, simply by virtue of the fact that they contain more vowels, and hence have more features to compare and potentially differ from others on. For this reason, it was useful to define distance as a per-word average. The simplest way to implement this is to compare only stem vowels: the first vowel in each of the words in the pairs. In this case (referred to as the stem vowels metric) it is unnecessary to correct for word length, as all words contributed equally to the overall distances. To compare more vowels per word, one possibility is to compare only as many vowel pairs as possible, matching up vowels one at a time, beginning at the first vowel of each word and omitting any vowels that did not match due to different word lengths or syllable number. In this case (referred to as the maximum vowel pairs metric), dividing a word pair's total number of feature differences by the number of vowel pairs examined made it possible to compare words of different lengths and combine their differences into a distance over the entire wordset. Finally, to compare the maximum number of vowels per word and add distance to mark the gain or loss of a syllable (and its vowel), a pretty significant evolutionary event, my approach was to compare as many vowel pairs as could be constructed, and add constants for all the leftover unpaired vowels. I used the maximum possible feature changes on a vowel for the constant in this metric (referred to in subsequent discussion as the maximum vowel pairs method with unmatched vowels marked). These three metrics

4.6.2 Loanwords

Testing word pairs phoneme for phoneme, vowel pair by vowel pair, feature by feature, seems like a good way of measuring linguistic evolution by sound changes in cognates, words descended from the same ancestral lexical item. But what about loanwords, which share the meaning of words descended genetically from ancestral languages, but were borrowed into the language as a result of contact with another? As mentioned earlier, borrowings could result in word pairs with mismatched syllable numbers. Loanwords also tend to look very different from cognates with the same meaning. To mark loanwords and the fact that they constitute a more significant evolutionary event than sound change, I had my distance finder test if word pairs matched on their initial phoneme and on vowel (syllable) number. If neither of these were true, it seemed likely that one of the words in the pair was a loan, and I had the program add the maximum possible sound change for that pair. For contrast, I also implemented versions of each distance finder that did not check for loanwords at all, much less account for them. The results of each are presented and discussed in Chapter 5.

4.6.3 Diphthongs

In addition to the problem of loanwords and mismatched vowel number, there was the question of diphthongs to resolve. Diphthongs are vowels made up of two vowel sounds, or a vowel and a glide. They can be classified according to the features of their glide, which in **onglides** begins the diphthong and in **offglides** ends it. The diphthongs of the languages I studied are shown in Table 4.2.

The question of how to encode diphthongs posed a serious problem for computing distances, especially as representing diphthongs as a sequence of two vowels seemed likely to create more word pairs with mismatched vowel lengths. So it was necessary to find a way to match and compare single vowels to diphthongs. One possibility was to use only the first vowel of each diphthong, a strategy similar to comparing word pairs only on root vowels, but was unattractive for the same reason: it overlooked a great deal of potentially informative data. Another was to average the features of the two vowels in each diphthong to create a combination vowel. I rejected this approach because by this reasoning the diphthong **au**, a combination of a low front unrounded vowel and a high back unrounded vowel, is equivalent to the mid central unrounded

	front	back
rounded	**offglides:** ey, əy, œy	**offglides:** iu, oʊ, ʌi
unrounded	**onglides:** ɪɛ, ɪɯ, ɪə, ɪo, ɪa **offglides:** ei, ɔi, ai, ʌi	*(none)*

Table 4.2: Diphthongs in English, Dutch, German, Icelandic, Norwegian, and Spanish.

model all vowels, including single vowels of all lengths, as diphthongs. There is some precedent for this: some theories of English phonology consider the long vowels diphthnongs [24]. In my formulation, regular length single vowels are a combination of their phoneme and a null vowel, which has no features, and as such differs by at least four sound changes from all other vowels (and a minimum of eight from diphthongs). In this system, long vowels are two regular-length vowels in a row, and diphthongs are a combination of their two component parts. (See Appendix D for all the encodings.)

Chapter 5

Results and Analysis of Distance-Based Methods

Three possible approaches to word length and syllable gain or loss (stem vowels, maximum vowel pairs, and maximum vowel pairs with unmatched vowels marked), and two possible treatments of loanwords (detect or ignore) made six possible distance matrices. As mentioned earlier, I used PHYLIP, an extensively documented free software phylogeny-construction package developed by researchers at the University of Washington [20], to run Fitch-Margoliash least-squares and neighbor-joining on these, for a total of twelve trees, presented in Section 5.1. Section 5.2 compares the results to a traditional historical tree for the Germanic languages and Spanish, studied in Section 5.2, and the rest of my analysis and observations are in Section 5.3.

5.1 A Forest of Results

The following are the trees generated from distance matrices based on 112 words (90 nouns, 22 adjectives) in the six languages I studied. The trees produced by neighbor-joining are in Section 5.1.1 and those from the least-squares method of Fitch and Margoliash in Section 5.1.2). Each tree is captioned with the measure used in producing the distance matrix from which it was generated.

5.1.1 Neighbor-Joining Method

Some observations about these results: Figures 5.1 and 5.2 are identical, as are Figures 5.28 and 5.5, and, most interestingly, Figures 5.3 and 5.6. The latter result suggests that testing for loanwords and adjusting distances to reflect them may not be informative. The vowel distances added by the loanwords without any additional weighting seem to have been enough in this

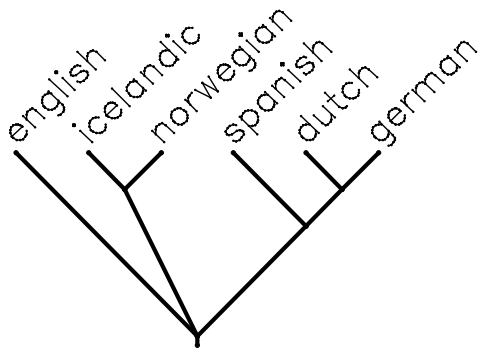


Figure 5.1: Neighbor method, stem vowels metric, loanwords marked.

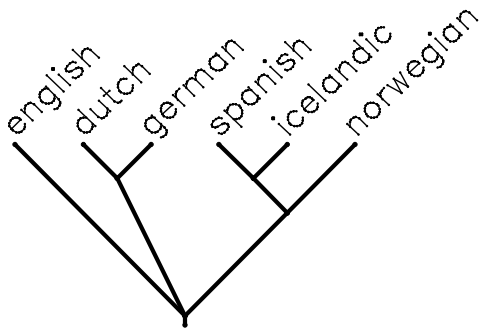


Figure 5.2: Neighbor method, maximum vowel pairs metric, loanwords marked.

5.1.2 Fitch-Margoliash Least Squares Method

A few notes about these trees as well: As in the trees discussed in Section 5.1.1, Dutch and German are always nearest neighbors (often siblings, in fact). Figures 5.8 and 5.9 are identical, as are Figures 5.10 and 5.11. But Figure 5.12 is the real prize of the group, as is revealed immediately by comparison to Figure 5.2.

5.2 Comparison To Historical Conclusions

It may seem redundant to have tested ditance methods on as well-known and intensively studied language family as Germanic. After all, the tree for its descendants is fairly well agreed-upon, even if its position in the larger Indo-European tree is less clear. On the other hand, studying a well-understood group offers the benefit of results against which to compare mine (see Figure 5.2). For testing methods of generating trees for languages based solely on contemporary

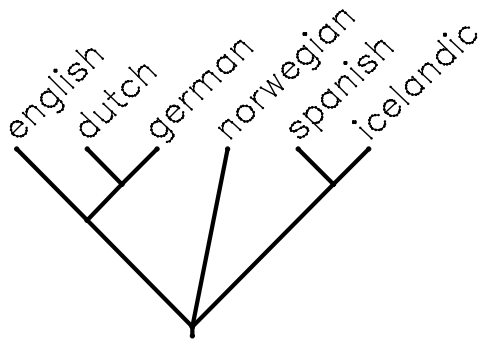


Figure 5.3: Neighbor method, maximum vowel pairs metric, loanwords and unpaired vowels marked.

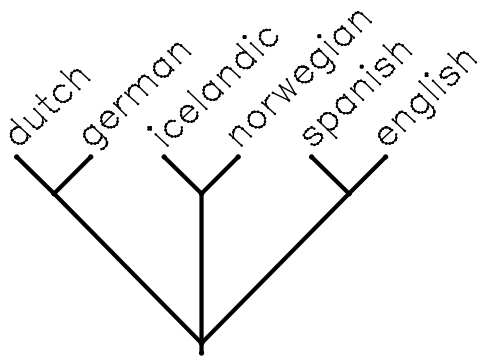


Figure 5.4: Neighbor method, stem vowels metric, no loanword checks.

suring. Depending on one's interpretation of the traditional family tree for the descendants of Indo-European, Figure 5.1.2 is just as valid a phylogeny for these six languages as as Figure 5.2. After all, both the Germanic family and the Romance family, of which Spanish is a member, descended directly from Indo-European.

5.3 Analysis and Observations

In addition to the close relationship observed between Dutch and German (and often English, who appeared adjacent to those frequent siblings in nine out of twelve trees). Similarly, Icelandic and Norwegian were siblings or adjacent nodes in all but three trees. So even when the distance methods did not generate the correct tree, they revealed relationships between the languages being studied. Consensus-tree methods such as those used by the CHLP might be able to tease out the correct tree given a group of results such as these [37].

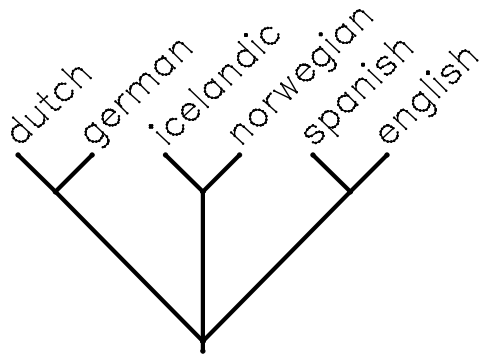


Figure 5.5: Neighbor method, maximum vowel pairs metric, no loanword checks.

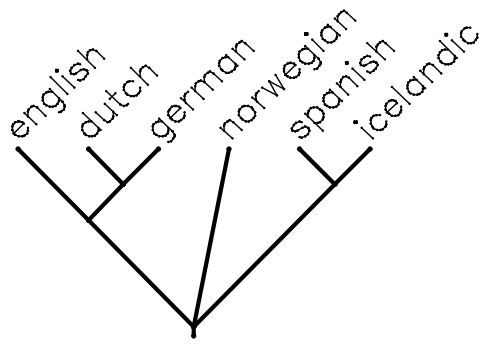


Figure 5.6: Neighbor method, aximum vowel pairs metric, unpaired vowels marked, no loanwords marked.

firm conclusions can be drawn. Although for the neighbor-joining method marking for loanwords did not seem to make much of a difference, the "correct" tree produced by the Fitch-Margolias method was under a language that did not mark for loanwords, and the equivalent distance matrix that included added distance for suspected loanwords did not produce the same result. Likewise, adding constants for unpaired vowels seemed to help, so it seems that the more the distance reflects the actual phonology of the situation, the better. This should come as no surprise, given the Farach-Kannan result that distance methods converge to a single best tree for the evolutionary model proposed given enough data [12].

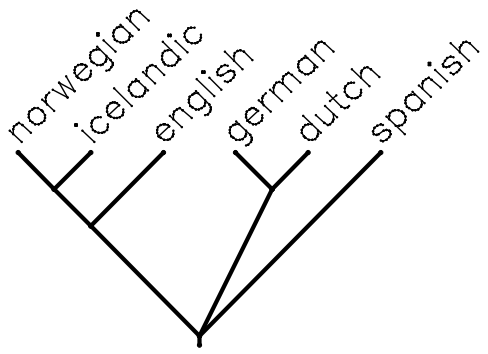


Figure 5.7: Fitch method, stem vowels metric, loanwords marked.

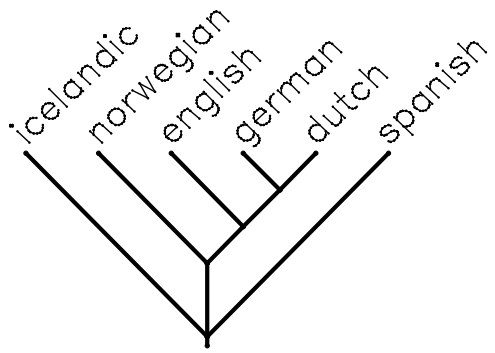


Figure 5.8: Fitch method, maximum pairs metric, loanwords marked.

File missing.

Figure 5.9: Fitch method, maximum pairs metric, loanwords and unpaired vowels marked.

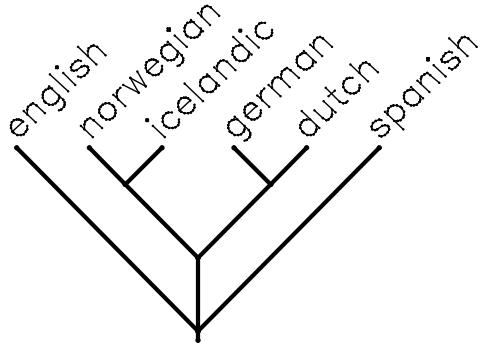


Figure 5.10: Fitch method, stem vowels metric, no loanword checks.

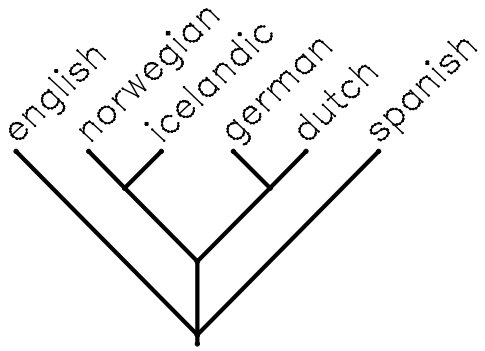


Figure 5.11: Fitch method, maximum vowel pairs metric, no loanword checks.

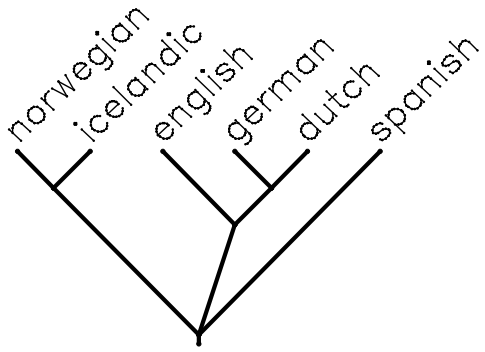


Figure 5.12: Fitch method, maximum vowel pairs metric, unpaired vowels marked, no loanwords marked.

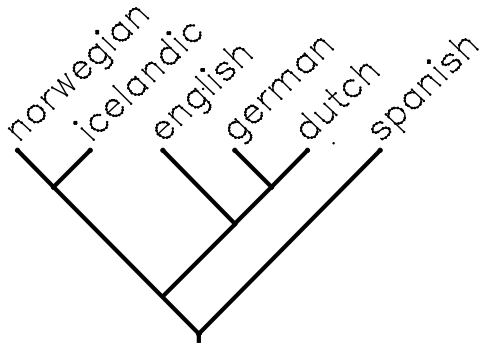


Figure 5.13: Historical tree for English, Dutch, German, Icelandic, Norwegian, and Spanish. Adapted from [42]

Chapter 6

Conclusion

On an autobiographical note, which I am wont to insert into everything I write, especially after several hours of staying up way too late working, the scope and ambition of this project finally occurred to me at a similarly inhuman hour less than two months ago. Of course by that point it was far too late to change my goals, and I was far too determined to accomplish some of what I'd set out to do. So I plowed ahead, and as a result accomplished both more and less than I expected. Section 6.1 of this conclusion addresses the former result; Section 6.2 the latter.

6.1 Summary of Results

The results presented in Chapter 5 support the hypothesis that phylogeny methods, and in particular adaptation of distance-based tree construction algorithms, can be used to deduce the evolutionary history of language families. Cool. It is especially gratifying to notice that certain relationships recurred even the trees which were quite dissimilar from the historical interpretation of the evolution of my language set: Dutch and German as closely related, even siblings, often related to English; and Icelandic and Norwegian as closely related, often sibling languages as well. So distance methods might be useful in detecting relationships between languages whose histories are not well-known. But before I start speculating on what else I might do with this project if given more time, let me discuss the conclusion that's likely to be of the most interest to historical linguists: vowels are historically informative! My trees were based almost entirely on vowel distances, and yet I managed to generate several accurate and relevant relationships between the languages I studied, including a correct phylogeny! That's so cool!

6.2 Avenues of Further Research

As I mentioned in Chapter 5, further testing is required before I draw any definitive conclusion about distance metrics. I would definitely implement a system similar to that for vowels to

method on other well-studied families, perhaps the Romance languages, since Spanish is already begun. If that were a success, I might attempt to address open problems in historical linguistics. An obvious place to go for more sources against which to test my conclusion would be the other group who used phylogenies on languages. As mentioned in Section 2.3, the CHLP based their lexical characters on a list of vocabulary from [47]. It would be interesting to construct a vocabulary list based on the same data they used and build distance trees to compare to their character-based phylogeny.

6.2.1 Alternative Distance Metrics

My distance metric rates vowels on an absolute 0-6 scale of how extreme the vowel with respect to the other possible values of its character. Another possibility is to develop a strictly phonemic scale, based on the phonologies of the languages being compared. In the case of English, Dutch, German, Icelandic, Norwegian, and Spanish, this would involve three levels of front-backness, six of height, two for rounding/unrounding, and between one and four or maybe five for length. Alternatively, I could implement a metric based on strictly binary feature-based models such as the ones described in [10] and [9], or perhaps on strict acoustic phonetic data, such as spectrograms. It would be interesting to see how each of these affected the distance matrices produced, and the trees generated from each.

6.2.2 Possible Extensions of Tree Models

As I mentioned in Chapter 3, no method of describing language change would be complete without a good way of accounting for the effects of language contact. Perhaps the most interesting open problem left available in the study of language change through computational biological methods is that of expanding existing models to describe language change, and in particular contact languages. To stay as close to the existing tree-based methods would probably require their extension into networks, which in turn might be used to detect horizontal gene transfer, the default explanation of poor tree-based results in studies of molecular phylogeny in simple organisms capable of exchanging genetic material through other than strictly hereditary mechanisms. If I had more time, I would probably approach this problem by attempting to draw an evolutionary tree for a well-studied contact language and its ancestors. Based on the results

and test the network models on known strictly-tree families often and keeping in mind that the formation of contact languages is a very rare occurrence.

Appendix A

Vocabulary List

As described in Section 4.3, I developed the following list of vocabulary to use as wordsets to compare and compute vowel distance matrices with the programs in Appendix B. The first column is the English for the lexical items represented phonetically by the IPA strings in the columns to the right. To create the word lists for each language used in the distance finders, the text of the LaTeX file for the data tables in this appendix were split into symbols for phonemes using Emacs regular expression operations and the modified LaTeX file (munge.tex) was divided using the Unix `cut` command (`cat munge.tex | cut -d & -f i > languagei.words` for the `i`th column of the file).

A.1 Nouns

A.1.1 Numbers (15)

word	**English**	**Dutch**	**German**	**Icelandic**	**Norwegian**	**Spanish**
one	wən	ʔeːn	ʔaɪn	ʔeɪtN	ʔe y n	uno
two	tuː	tweː	tsvaɪ	tveɪr	tuː	dos
three	thriː	driː	draɪ	thriːr	trɛt	tres
four	fɔr	viəˑ	fiəˑ	fjøɢʏR	firɪ	kwᴀtro
five	faɪv	væɪf	fʏnf	fɪmm	fɛ	thingko
six	sɪks	zɛs	zɛks	sɛks	sɛks	seɪs
seven	sɛvn̩	zevn̩	ziːbən	sjø	ʃy	sjete

Numbers, cont.

word	English	Dutch	German	Icelandic	Norwegian	Spanish
eight	ʔeit	ʔaχt	ʔaχt	ʔaʊta	ʔottə	otʃo
nine	naɪn	neɢn̩	noɪn	niːʏ	niː	nweve
ten	tɛ	tiːn	tseːn	tiːʏ	tiː	djeθ
seventeen	sɛvn̩tiːn	zevn̩tiːn	ziːptsen	seɪtjaʊn	sʏtn̩	djeθisjete
twenty	twɛti	twɪntɪɢ	tsvantsɪχ	tʏtʏgʏ	tivɛ	beɪnte
thirty	thəˑti	dɛrtɪɢ	draɪsiχ	θrjaʊtiːʏ	trɛtiː	treɪntᴀ
forty-two	fɔrtituː	tweenn̩fɪrtɪɢ	tsvaɪʔʊntfiːrtsɪχ	fjørtiːoɢtveɪr	frtituː	kwᴀrentaɪdos
hundred	həndrɛd	hɔndəˑd	hʊndəˑt	hʏndrath	hyndrœ	θjen

A.1.2 Time (12)

word	English	Dutch	German	Icelandic	Norwegian	Spanish
time	tᴀɪm	taɪt	tsaɪt	tiːmi	tiːd	tjɛmpo
year	jɪəˑ	jaːr	jaːr	aʊr	oəˑ	año
month	mənth	maant	moːnat	maʊnuthəˑ	monʏd	mes
week	wiːk	veːk	vɔχə	viːka	ʔykɛ	semana
day	deɪ	dɑχ	tak	daɢʏR	dɔg	dia
hour	ʔau wəˑ	ʔyːr	ʃtʊndə	klʏhkʏstʏnd	tiːmə	ora
minute	mɪnət	minyt	miːnuːtə	miːnʊta	mɪnyt	minuto
second	sɛkənd	sœkɔndœ	zɛkʊndə	sɛkunda	dɛkynd	segundo
morning	mɔrnɪng	mɔrgn̩	mɔrgən	mɔrgʊn	mɔrɛn	mañana
night	nəɪt	nɑχt	naχt	noːt	nat	notʃe
afternoon	ʔæftəˑnuːn	mɪdɑχ	naχmɪtak	ʔɛftɪrmɪthdaɢʏR	ʔɛtmidag	tarde
evening	ʔivnɪng	ʔavɔnt	aːbənt	kvølt	kvɛl	tarde

A.1.3 Nature (15)

word	English	Dutch	German	Icelandic	Norwegian	Spanish
sun	sən	zɔn	zɔnə	soːl	syl	sol
rain	rein	reχn̩	reːgən	rɛgn	reɪn	juvja
snow	snoʊ	sneːy	ʃneː	sñoːr	snø	njeve

Nature, cont.

word	English	Dutch	German	Icelandic	Norwegian	Spanish
cloud	klaʊd	wɔlk	vɔlkə	skiː	ʃiː	nube
sky	skaɪ	lœχt	hɪməl	loft	himmɛ	θjelo
moon	muːn	maːn	moːnt	tunggl̩	monɛ	luna
star	star	stɛr	ʃtɛrn	stjartna	stjɛrnɛ	estreja
planet	plænət	planeːt	planeːt	reɪkɪstjartna	planɛt	planeta
winter	wɪntəɾ	vɪntəɾ	vɪntəɾ	vɛtʏr	vintəɾ	invjerno
spring	sprɪng	vɔrjaar	frylɪng	vɔrlɛk	vɔr	primavera
summer	səməɾ	zoməɾ	zɔməɾ	sʏmaR	sɔmɛr	verano
autumn	ʔatm̩	hɛrfst	hɛəɾpst	høyst	hʊst	otoño
storm	stɔːrm	stɔrm	sturm	stɔrmʏR	stɔrm	tormenta
fire	faɪjəɾ	vyːr	foijəɾ	ɛldʊr	bran	fwego
water	watəɾ	vatəɾ	vasəɾ	vatn̩	wan	agwa

A.1.4 Geography(18)

word	English	Dutch	German	Icelandic	Norwegian	Spanish
mountain	mauntn̩	bɛr χ	bɛrk	fjatl	fjɛl	montaña
lake	leɪk	mɪr	ze	sjoor	inʃʏ	lago
river	rɪvəɾ	rəɾviɛr	flʊs	fljoot	ɛlv	rio
ocean	ʔoʃn̩	ʔoseaan	oʒeaːn	haf	havɛ	oθean
sea	siː	zeː	mer	haf	ʃø	mar
land	lænd	lant	lant	lant	lan	tjera
earth	ʔəɾθ	ʔaardɛ	ʔɛrdə	ɪørθ	jyɪr	tjera
ground	graʊnd	gʁɔnd	grʊnt	gʁʏnd	bakɪn	tjera
hill	hɪl	hɛʊvl̩	hygəl	hotl	hai	kolina
pond	pɔnd	plɔs	teiχ	tjørdN	wan	estanke
forest	fʊrɪst	bɔs	fɔrst	skoʊwʏR	skyg	boske
woods	wʊds	waʊd	valt	skoʊwʏR	skyg	boske
stream	striːm	beːk	baχ	strəːm	bɛk	arojo
island	ʔaɪləɴd	ʔæiləɴd	ʔɪnsəl	ʔeya	ʔəʏ	isla

Geography, cont.

word	English	Dutch	German	Icelandic	Norwegian	Spanish
north	nɔrθth	nɔrd	nɔrdən	nɔthʏR	nyʀ	norte
south	sʌʊth	zayd	zydən	sʏthʏR	syd	sur
east	ʔiːst	ʔoːst	ʔoːstən	ʔʌʊstʏR	ʔəʏst	este
west	wɛst	vɛst	vɛstən	vɛstʏR	vɛst	oeste

A.1.5 Animals(16)

word	English	Dutch	German	Icelandic	Norwegian	Spanish
cat	kæt	kɑt	katsə	køhtyʀ	kat	gato
dog	dɔg	hɔnd	hʊnt	hynduʀ	hynd	pero
mouse	maʊs	məɪs	maʊs	mus	mys	raton
rat	ræt	rɑt	ratə	rɔta	rɔtɪ	rata
pig	pɪg	vʌɪkn̩	ʃvain	svin	grɪs	θerdo
cow	kaʊ	kuː	kuː	kir	ky	baka
livestock	laɪvstɔk	veː	fi	bupenɪngyʀ	fjʊstɪʀ	ganado
horse	hɔrs	paːrd	pfɛrt	hɛstyʀ	hɛ	kabajo
chicken	tʃɪkn̩	kɪp	kykχən	kjuhklɪnggyʀ	hœnɛ	pojo
goat	gɔt	xæit	gais	geɪt	jeɪt	kabra
sheep	ʃiːp	sχaːp	ʃaːf	saʊthkɪnt	səy	obeχa
bird	bɚd	voχl̩	foːgəl	fʊkL	fyl	paχaro
duck	dək	ʔeːnt	ʔɛntə	ənd	ʔan	pato
goose	guːs	χɑns	gʌns	gaɪs	gos	oka
swan	swʌn	zvaːn	ʃvaː	svanʏR	swʌnɛ	θisne
fish	fɪʃ	vɪs	fɪʃ	fɪskʊR	fɪsk	peθ

A.1.6 People (8)

word	English	Dutch	German	Icelandic	Norwegian	Spanish
woman	wʊmn̩	vrau	frau	kɔna	kvinɛ	muχer
man	mæn	mɑn	man	mʌthʊr	man	ombre

word	English	Dutch	German	Icelandic	Norwegian	Spanish
child	tjaild	kɪnt	kɪnt	barn	bʌrn	niño
girl	gəˠl	mɛiʃœ	meːtχn̩	stulka	jɛrɴ	niña
boy	bɔɪ	jɔngn̩	jʊngə	dreɪngʊʀ	gʊt	niño
adult	ʔədl̩t	vɔlwɑsn̩	ʔɛvaksənəˠ	vʌksɪn	vɔlksn̩	adulto
student	studn̩t	lɪɪrlɪng	ʃtudɛnt	nemandɪ	stydɛnt	estudjante

A.1.7 Kinship Terms (6)

word	English	Dutch	German	Icelandic	Norwegian	Spanish
mother	məthəˠ	mudəˠ	mʊttəˠ	mothɪʀ	moːəˠ	madre
father	faθəˠ	vɑdəˠ	vɑtəˠ	fathɪʀ	fær	padre
daughter	dɔtəˠ	dɔxtəˠ	tɔχtəˠ	dohtɪʀ	dottɪʀ	iχa
son	sən	zoːn	zoːn	sɔnːuʀ	dən	iχo
sister	sɪstəˠ	zʏs	ʃvɛstəˠ	sɪstɪʀ	səstɪʀ	ermana
brother	brəθəˠ	bruəˠ	brudəˠː	brothɪʀ	bruəˠ	ermano

A.2 Adjectives (22)

word	English	Dutch	German	Icelandic	Norwegian	Spanish
tired	tʌɪjəˠd	mu	myːdə	threɪtʊR	trət	kansado
hungry	həngriː	hɔngəˠɪχ	hʊngrχχ	hʊgrʌthʊR	sytn̩	ambrɪento
hot	hʌt	heːt	haɪs	heɪtʊR	warm	kaljente
cold	kɔld	kaʊt	kalt	kaldʊr	kalt	frio
warm	wʌrm	vɑrm	vɑrm	varmʊr	wʌrm	templado
cool	kuːl	kul	kyːl	svalʊr	hjœlɪ	fresko
big	bɪg	χroːt	groːs	stoːr	stoːəˠ	grande
little	lɪɾl̩	klaɪn	klein	liːtɪλ	litɛn	pekeño
small	wəɪd	waɪd	braɪt	vithʊr	breɪviː	antʃo
broad	brɔd	breːd	braɪt	breɪthʊr	breɪviː	antʃo
thin	thɪn	dœn	dʏn	thʊnnʊr	tiː	delgado

Adjectives, cont.

word	English	Dutch	German	Icelandic	Norwegian	Spanish
thick	thɪk	dɪk	dɪk	thʊkkʊr	tək	grueso
tall	təl	lʌng	groːs	har	hœy	alto
short	ʃɔrt	kɔrt	kʊts	stuttʊr	kɔrt	korto
long	lɔng	lɑng	ɑ	langʊr	lang	largo
good	gʊd	χut	gut	gɔt	bra	bweno
bad	bæd	slɛɪt	ʃlɛχt	vondʊ	dɔrlɛ	malo
fat	fæt	vɛt	fɛt	feɪttʊr	fyk	gɔrdo
clean	kliːn	sχoːn	zaʊbəʳ	hreɪn	rien	limpjo
dirty	dəʳdi	vəyl	ʃmʊtsɪχ	ohreɪn	ʃidɛn	suθio
high	haɪj	hoː	hoːχ	haʊr	hœy	alto
low	loʊ	laː	nidrɪχ	lagʊr	lʌv	baχo

Appendix B

Source Code

The following Python code implements the various vowel distance measures described in Section 4.6.1. Priscilla is a vehicle for Bernadette, Mitzi, Felicia, Ralph, Teek, or Adam, which it stores as libraries and calls to do pairwise comparisons between all the languages being studied and produce a distance matrix accordingly. The distance finders share the library Shoe, which stores several functions most of them have in common.

B.1 Priscilla.py

Constructs and prints out a matrix of pairwise vowel distances between the word sets for the six languages studied (see Appendix A for data). The distances are computed by calls to the `compare.language` function of any of bernadette.py, ralph.py, mitzi.py, teek.py, or felicia.py (Sections B.3, B.4 B.5, B.6, and B.7, respectively).

```
#! /usr/bin/python
import sys
import shoe
import bernadette
import mitzi
import felicia
import ralph
import teek
import adam

languagenames = ['spanish',\
                 'english',\
                 'dutch',\
```

```
                ’icelandic’,\
                ’norwegian’]

vowels = shoe.readsounds(’vowels.txt’)
consonants = shoe.readsounds(’consonants.txt’)

language = []

for i in range(len(languagenames)):
    language.append(shoe.readlang(languagenames[i] + ’.words’))

print len(languagenames)

for i in range(len(languagenames)):
    print languagenames[i],
    for k in range(10-len(languagenames[i])):
        print " ",
    for j in range(len(languagenames)):

        if sys.argv[1] == ’1’:
            diff = bernadette.comparelanguage(language[i],\
                                              language[j],\
                                              vowels,\
                                              consonants)
        elif sys.argv[1] == ’2’:
            diff = mitzi.comparelanguage(language[i],\
                                         language[j],\
                                         vowels,\
                                         consonants)

        elif sys.argv[1] == ’3’:
```

```
                                        language[j],\
                                        vowels,\
                                        consonants)

        elif sys.argv[1] == '4':
            diff = ralph.comparelanguage(language[i],\
                                         language[j],\
                                         vowels,\
                                         consonants)

        elif sys.argv[1] == '5':
            diff = teek.comparelanguage(language[i],\
                                        language[j],\
                                        vowels,\
                                        consonants)
        elif sys.argv[1] == '6':
            diff = adam.comparelanguage(language[i],\
                                        language[j],
                                        vowels,\
                                        consonants)

        print diff,
    print ''
```

B.2 Shoe.py

Shoe is a library of functions for common use in all the distance finders: reading in wordsets for languages, reading in vowel names and features to create vowel and consonant dictionaries, transforming a word into a string of vowels, comparing pairs of vowel strings vowel by vowel, and comparing individual vowels feature by feature.

```
#! /usr/bin/python
```

```
def comparevowels(dutchvowels, englishvowels, voweldict):
    """compares pairwise the feature values for two vowel sets of
    equal length (one for each language) and returns a distance
    between the two"""

    if len(dutchvowels) != len(englishvowels):
        print 'cannot compare two vowel strings of different length'
        return -1

    distance = 0
    for i in range(len(dutchvowels)):
        distance = distance + comparevowel(dutchvowels[i],\
                                           englishvowels[i],\
                                           voweldict)

    return float(distance)

def comparevowel(dutchvowel, englishvowel, voweldict):
    """compares a pair of vowels by translating them into their
    feature vectors and returns distance between them: the number of
    changes required to transform one into the other"""

    dutchfeatures = voweldict[dutchvowel]
    englishfeatures = voweldict[englishvowel]

    difference = 0

    for i in range(len(dutchfeatures)):
        difference = difference + abs(dutchfeatures[i] - englishfeatures[i])
```

```
def vowelize(word, vowels, consonants):
    "returns all the vowels of a word (strips out consonants)"

    wordvowels = []
    for sound in word:
        if vowels.has_key(sound):
            wordvowels.append(sound)
        elif consonants.has_key(sound):
            pass
        else:
            print 'not in vowel or consonant dictionary: ', sound

    return wordvowels

def readlang(filename):
    "reads in a language from filename and returns an array (words)"

    file = open(filename, 'r')
    lines = file.readlines()

    words = []

    for line in lines:
        words.append(string.split(line))

    return words

def readsounds(filename):
    """reads in list of sound symbols and corresponding features from
    filename and returns a vowel dictionary (sounds) that maps symbols
```

```
    file = open(filename, 'r')
    file.readline()
    lines = file.readlines()

    sounds = {}

    for line in lines:
        stuff = string.split(line)
        sound = stuff[0]
        features = []

        for i in range(1, len(stuff)):
            features.append(string.atoi(stuff[i]))

        sounds[sound] = features

    return sounds
```

B.3 Bernadette.py

The following program prints out the vowel distance between two languages based only on root vowel comparisons of words that appeared to be cognates (adding the maximum possible distance for probable loanwords, defined as word pairs with mismatched vowel number **and** initial phoneme.

```
#! /usr/bin/python
import string
import sys
import shoe
```

```
def comparelanguage(dutch, english, vowels, consonants):
    """computes the average phonemic distance between two wordsets as
    the mean difference between the stem vowels of all word pairs."""

    if len(dutch) != len(english):
        print 'ack! word sets of different length! cannot compare!'
        return -666

    distance = 0

    for i in range(len(dutch)):
        distance = distance + compareword(dutch[i],\
                                          english[i],\
                                          vowels,\
                                          consonants)

        avgdistance = float(distance)/len(dutch)

    return avgdistance

def compareword(dutchword, englishword, voweldict, consdict):
    """compares two words (checks if first consonants and vowel
    numbers match, since those are likely to indicate loanwords and/or
    insertion or deletion of syllables) and passes stem vowels to
    comparevowel to compute distance, which is returned"""

    distance = 0

    if englishword[0] != dutchword[0]:
        distance = distance + 1
```

```
    dutchvowels = shoe.vowelize(dutchword, voweldict, consdict)
    englishvowels = shoe.vowelize(englishword, voweldict, consdict)

    if len(dutchvowels) != len(englishvowels):
        distance = distance + 1

    if distance < 2:

        distance = float(distance)/48 + shoe.comparevowel(dutchvowels[0],\
                                                          englishvowels[0],\
                                                          voweldict)
    else:
        distance = 1

    return distance

if __name__ == '__main__':
    dutch = shoe.readlang(sys.argv[1])
    english = shoe.readlang(sys.argv[2])
    vowels = shoe.readsounds(sys.argv[3])
    consonants = shoe.readsounds(sys.argv[4])

    print 'distance: ', comparelanguage(dutch, english, vowels, consonants)
```

B.4 Ralph.py

This program prints out the vowel distance between two languages based on root vowel comparisons without checking for loanwords: all wordpairs are treated as cognates.

```
#! /usr/bin/python
import string
import sys
```

```
def comparelanguage(dutch, english, vowels, consonants):
    """computes the average phonemic distance between two wordsets as
    the mean difference between the stem vowels of all word pairs."""

    if len(dutch) != len(english):
        print 'ack! word sets of different length! cannot compare!'
        return -666

    distance = 0

    for i in range(len(dutch)):
        distance = distance + compareword(dutch[i],\
                                          english[i],\
                                          vowels,\
                                          consonants)
        avgdistance = float(distance)/len(dutch)

    return avgdistance

def compareword(dutchword, englishword, voweldict, consdict):
    """compares two words by vowelizing and passing stem vowels to
    comparevowel to compute distance, which is returned"""

    distance = 0

    dutchvowels = shoe.vowelize(dutchword, voweldict, consdict)
    englishvowels = shoe.vowelize(englishword, voweldict, consdict)

    distance = distance + shoe.comparevowel(dutchvowels[0],\
         englishvowels[0],\
```

```
    distance = float(distance)
    return distance

if __name__ == '__main__':
    dutch = shoe.readlang(sys.argv[1])
    english = shoe.readlang(sys.argv[2])
    vowels = shoe.readsounds(sys.argv[3])
    consonants = shoe.readsounds(sys.argv[4])

    print 'distance: ', comparelanguage(dutch, english, vowels, consonants)
```

B.5 Mitzi.py

Mitzi prints out the vowel distance between two languages based on as many pairwise comparisons as possible, checking cognation by examining initial phoneme and word length.

```
#! /usr/bin/python
import string
import sys
import shoe

def comparelanguage(dutch, english, vowels, consonants):
    """computes the average vowel distance between two wordsets by
    summing the vowel distances between pairs of cognates (maximum
    distance between probable loanwords) and dividing by the number of
    pairs"""

    if len(dutch) != len(english):
        print 'ack! word sets of different length! cannot compare!'
        return -666
```

```
    for i in range(len(dutch)):
        distance = distance + compareword(dutch[i],\
                                          english[i],\
                                          vowels,\
                                          consonants)


        avgdistance = float(distance)/len(dutch)
    return avgdistance

def compareword(dutchword, englishword, voweldict, consdict):
    """compares two words: vowelizes, passes to compareword if vowel
    numbers match, otherwise checks if initial consonants match and
    adds the maximum possible distance if they don't; all other vowel
    strings of mismatched length are cropped to matched lengths and
    passed through compareword to compute distance, which is
    returned."""

    distance = 0

    dutchvowels = shoe.vowelize(dutchword, voweldict, consdict)
    englishvowels = shoe.vowelize(englishword, voweldict, consdict)

    if len(dutchvowels) == len(englishvowels):
        distance = shoe.comparevowels(dutchvowels, englishvowels, voweldict)

    elif dutchvowels[0] != englishvowels[0]:
        distance = 1

    else:
```

```
        distance = shoe.comparevowels(dutchvowels[:maxpairs],\
                                      englishvowels[:maxpairs],\
                                      voweldict)

    return float(distance)

if __name__ == '__main__':
    dutch = shoe.readlang(sys.argv[1])
    english = shoe.readlang(sys.argv[2])
    vowels = shoe.readsounds(sys.argv[3])
    consonants = shoe.readsounds(sys.argv[4])

    print 'distance: ', comparelanguage(dutch, english, vowels, consonants)
```

B.6 Teek.py

Teek prints out the vowel distance between two languages generated by comparing as many vowel pairs per word as possible, under the assumption that all word pairs are cognates.

```
#! /usr/bin/python
import string
import sys
import shoe

def comparelanguage(dutch, english, vowels, consonants):
    """computes average vowel distance between two wordsets by
    summing vowel distances between word pairs (assumed to be
    cognates) and dividing by the number of pairs"""

    if len(dutch) != len(english):
        print 'ack! word sets of different length! cannot compare!'
```

```
    distance = 0

    for i in range(len(dutch)):
        distance = distance + compareword(dutch[i],\
  english[i],\
  vowels,\
  consonants)

        avgdistance = float(distance)/len(dutch)
    return avgdistance

def compareword(dutchword, englishword, voweldict, consdict):
    """compares two words as if they were cognates: vowelizes, crops
    vowel strings to match lengths, and passes them through
    compareword to compute distance, which is returned."""

    distance = 0

    dutchvowels = shoe.vowelize(dutchword, voweldict, consdict)
    englishvowels = shoe.vowelize(englishword, voweldict, consdict)

    if len(dutchvowels) == len(englishvowels):
        distance = shoe.comparevowels(dutchvowels, englishvowels, voweldict)

    else:
        maxpairs = min(len(dutchvowels), len(englishvowels))

        distance = shoe.comparevowels(dutchvowels[:maxpairs],\
      englishvowels[:maxpairs],\
      voweldict)
```

```
    return float(distance)

if __name__ == '__main__':
    dutch = shoe.readlang(sys.argv[1])
    english = shoe.readlang(sys.argv[2])
    vowels = shoe.readsounds(sys.argv[3])
    consonants = shoe.readsounds(sys.argv[4])

    print 'distance: ', comparelanguage(dutch, english, vowels, consonants)
```

B.7 Felicia.py

Felicia prints out the vowel distances between two languages generated by comparing as many vowel pairs per word as possible, as Mitzi would, and adding distance for each unpaired vowel: its comparison with the null vowel (the zero feature vector). This added distance indicates that a syllable has been lost or gained. Likely loanwords are detected and dealt with as in Bernadette and Mitzi, except the constant added is the maximum possible change for the longer of the two words.

```
#! /usr/bin/python
import string
import sys
import shoe

def comparelanguage(dutch, english, vowels, consonants):
    """computes the average distance between two wordsets by summing
    the distances between pairs of cognates and dividing by the number
    of pairs"""

    if len(dutch) != len(english):
        print 'ack! word sets of different length! cannot compare!'
        return -666
```

```
    distance = 0

    for i in range(len(dutch)):
        distance = distance +\
                   compareword(dutch[i], english[i], vowels, consonants)


        avgdistance = float(distance)/len(dutch)
    return avgdistance

def compareword(dutchword, englishword, voweldict, consdict):
    """compares vowel sets of two words (tests for loanwords, adds
    constant for mismatched vowel number, since it's likely to
    indicate insertion or deletion of syllables) and passes vowel sets
    through comparevowels to compute distance, which is returned"""

    dutchvowels = shoe.vowelize(dutchword, voweldict, consdict)
    englishvowels = shoe.vowelize(englishword, voweldict, consdict)

    distance = 0

    if dutchvowels[0] == englishvowels[0]:

        if len(dutchvowels) == len(englishvowels):
            longer = len(dutchvowels)
            distance = shoe.comparevowels(dutchvowels,\
                                          englishvowels,\
                                          voweldict)

            avgdist = float(distance)/longer
```

```
            long = dutchvowels
            maxpairs = len(englishvowels)
            longer = len(dutchvowels)

            distance = shoe.comparevowels(dutchvowels[:maxpairs],\
                                          englishvowels[:maxpairs],\
                                          voweldict)

            for i in range(longer-maxpairs):
                distance = distance + 1

            avgdist = distance/longer

        else:
            long = englishvowels
            maxpairs = len(dutchvowels)
            longer = len(englishvowels)

            distance = shoe.comparevowels(dutchvowels[:maxpairs],\
                                          englishvowels[:maxpairs],\
                                          voweldict)

            for i in range(longer-maxpairs):
                distance = distance + 1

            avgdist = float(distance)/longer

    else:
        if len(dutchvowels) == len(englishvowels):
            longer = len(dutchvowels)
            distance = shoe.comparevowels(dutchvowels,\
```

```
                                            voweldict)
            avgdist = float(distance)/longer

        else:
            avgdist = 1

    return avgdist

if __name__ == '__main__':
    dutch = shoe.readlang(sys.argv[1])
    english = shoe.readlang(sys.argv[2])
    vowels = shoe.readsounds(sys.argv[3])
    consonants = shoe.readsounds(sys.argv[4])

    print 'distance: ', comparelanguage(dutch,\
                                        english,\
                                        vowels,\
                                        consonants)
```

B.8 Adam.py

Adam is Felicia without the loanword checker.

```
#! /usr/bin/python
import string
import sys
import shoe

def comparelanguage(dutch, english, vowels, consonants):
    """computes the average distance between two wordsets by summing
    the distances between pairs of cognates and dividing by the number
    of pairs"""
```

```
    if len(dutch) != len(english):
        print 'ack! word sets of different length! cannot compare!'
        return -666

    distance = 0

    for i in range(len(dutch)):
        distance = distance +\
                   compareword(dutch[i], english[i], vowels, consonants)


        avgdistance = float(distance)/len(dutch)
    return avgdistance

def compareword(dutchword, englishword, voweldict, consdict):
    """compares two words (aligns vowelsets if necessary, adds
    constant for mismatched vowel number, since it's likely to
    indicate insertion or deletion of syllables) and passes aligned
    vowel sets through comparevowels to compute distance, which is
    returned"""

    dutchvowels = shoe.vowelize(dutchword, voweldict, consdict)
    englishvowels = shoe.vowelize(englishword, voweldict, consdict)

    distance = 0

    if len(dutchvowels) == len(englishvowels):
        longer = len(dutchvowels)
        distance = shoe.comparevowels(dutchvowels, englishvowels, voweldict)

    elif len(dutchvowels) > len(englishvowels):
```

```
        maxpairs = len(englishvowels)
        longer = len(dutchvowels)

        distance = shoe.comparevowels(dutchvowels[:maxpairs],\
                                      englishvowels[:maxpairs],\
                                      voweldict)

        for i in range(longer-maxpairs):
            distance = distance + shoe.comparevowel(long[maxpairs:][i],\
                                                    'null',\
                                                    voweldict)

    else:
        long = englishvowels
        maxpairs = len(dutchvowels)
        longer = len(englishvowels)

        distance = shoe.comparevowels(dutchvowels[:maxpairs],\
                                      englishvowels[:maxpairs],\
                                      voweldict)

        for i in range(longer-maxpairs):
            distance = distance + 1

    return distance/longer

if __name__ == '__main__':
    dutch = shoe.readlang(sys.argv[1])
    english = shoe.readlang(sys.argv[2])
    vowels = shoe.readsounds(sys.argv[3])
    consonants = shoe.readsounds(sys.argv[4])
```

```
print ’distance: ’, comparelanguage(dutch,\
                                   english,\
                                   vowels,\
                                   consonants)
```

Appendix C

Feature Dictionaries

The following dictionaries encode all of the sounds used in the vocabulary list of Appendix B as vectors of their features. Although only the vowels were compared in computing distances, it would be easy to expand the code in Appendix C and consonant feature vectors of Section C.2 to compute vowel distances that include consonant feature comparisons as well.

C.1 Vowels

The following are the vectors of feature values for the vowels of English, Dutch, German, Icelandic, Norwegian, and Spanish as encoded in Appendix B and used in the vowel dictionaries of the Python programs in Appendix C. Vowels heights ranged from 0 to 6, front to back distances were 3 to 0, rounding was either 2 or 1, and lengths ranged from 0 in the null vowel to 1 in reduced syllables such as the schwas (ə) to 2 in a regular syllable to 4 or sometimes 5 or 6 in double vowels or dipthongs. The null vowel (used in encoding regular vowels as dipthongs) is a string of zeros, and hence at least 4 changes away from any other vowel used in the second half of a dipthong or double vowel.

```
# symbol frontness height length rounding frontness height length rounding
null 0 0 0 0 0 0 0 0
i 6 6 6 0 0 0 0 0
i\textlengthmark 6 6 6 0 6 6 6 0
i\textrhookschwa 6 6 6 0 2 3 4 0
i\textepsilon 6 6 6 0 4 4 6 0
ie 6 6 6 0 5 4 6 0
io 6 6 6 0 0 3 6 6
iu 6 6 6 0 0 5 6 6
y 6 6 6 6 0 0 0 0
```

```
y\textsci 6 6 6 6 6 5 6 0
\textsci 6 5 6 0 0 0 0 0
\textsci\textsci 6 5 6 0 6 5 6 0
\textsci\textrhookschwa 6 5 6 0 2 3 4 0
\textsci\o 6 5 6 0 5 4 6 6
\textscy 6 5 6 6 0 0 0 0
e 5 4 6 0 0 0 0 0
ee 5 4 6 0 5 4 6 0
eea 5 4 6 0 2 2 6 0
eaa 5 4 6 0 2 2 6 0
ea\textlengthmark 5 4 6 0 2 2 6 0
ei 5 4 6 0 6 6 6 0
e\textsci 5 4 6 0 6 5 6 0
e\textlengthmark 5 4 6 0 5 4 6 0
e\textupsilon 5 4 6 0 6 5 6 0
e\textlengthmarky 5 4 4 0 6 6 6 6
ey 5 4 6 0 6 6 6 6
\o 5 4 6 6 0 0 0 0
\textepsilon 4 4 6 0 0 0 0 0
\oe 4 4 6 6 0 0 0 0
\oey 4 4 6 6 6 6 6 6
\textschwa 2 3 4 0 0 0 0 0
\textschwa\textscy 2 3 4 0 6 5 6 6
\textschway 2 3 4 0 6 6 6 6
\textschwa\textsci 2 3 4 0 6 5 6 0
\textschwa\textlengthmark 2 3 4 0 2 3 4 0
\textrhookschwa\textlengthmark 2 3 4 0 2 3 4 0
\textsyllabic{n} 2 3 3 0 0 0 0 0
\textsyllabic{m} 2 3 3 0 0 0 0 0
\textsyllabic{r} 2 3 2 0 0 0 0 0
\textsyllabic{l} 2 3 2 0 0 0 0 0
```

```
\ae 3 2 6 0 0 0 0 0
\ae\textsci 3 2 6 0 6 5 6 0
\aei 3 2 6 0 6 6 6 0
a 2 2 6 0 0 0 0 0
ai 2 2 6 0 6 6 6 0
aa 2 2 6 0 2 2 6 0
a\textlengthmark 2 2 6 0 2 2 6 0
a\textsci 2 2 6 0 6 5 6 0
a\textupsilon 2 2 6 0 0 4 5 6
au 2 2 6 0 0 5 6 6
\textbari 3 5 6 0 0 0 0 0
\textsca 3 0 6 0 0 0 0 0
\textsca\textupsilon 3 0 6 0 0 4 5 6
\textsca\textsci 3 0 6 0 6 5 6 0
\textscripta 1 0 6 0 0 0 0 0
\textscripta\textupsilon 1 0 6 0 0 4 5 6
\textopeno 0 2 5 6 0 0 0 0
\textopeno\textlengthmark 0 2 5 6 0 2 5 3
\textopeno\textsci 0 2 5 6 2 2 6 0
u 0 5 6 6 0 0 0 0
ue 0 5 6 6 5 4 6 0
u\textlengthmark 0 5 6 6 0 5 6 6
u\textrhookschwa 0 5 6 6 2 3 4 0
\textupsilon 0 4 5 6 0 0 0 0
o 0 3 6 6 0 0 0 0
o\textlengthmark 0 3 6 6 0 3 6 6
oo 0 3 6 6 0 3 6 6
o\textupsilon 0 3 6 6 0 4 5 6
o\textsci 0 3 6 6 6 5 6 0
oi 0 3 6 6 6 6 6 0
```

C.2 Consonants

To avoid accidentally overlooking any vowels not encoded in Section C.1, the following dictionary of consonants was made. Very few features were encoded, as its only purpose was to make sure all sounds were encoded and handled appropriately, but, as mentioned before, it would be easy to expand these feature vectors and adapt the distance finders to include additional data about consonants.

```
# symbol voiced sonorant
b 1 0
p 0 0
d 1 0
t 0 0
\textfishhookr 1 1
v 1 0
f 0 0
g 1 0
\textgamma 1 0
\textchi 0 0
x 1 0
k 0 0
\textscg 0 0
h 0 0
j 0 0
l 1 1
L 0 1
m 1 1
n 1 1
N 0 1
\textscn 0 1
\~n 1 1
ng 1 1
```

```
R 0 1
\textturnr 1 1
\textscr 0 1
\textinvscr 0 1
s 0 0
\textesh 0 0
\texttheta 0 0
t\textesh 0 0
th 1 0
w 1 1
\textglotstop 0 0
z 1 0
\textyogh 1 0
```

Bibliography

[1] R. Agarwala and D. Fernández-Baca. A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed. *SIAM Journal of Computing*, pages 1216–1224, 1994.

[2] R. Appel and P. Muysken. *Language Contact and Bilingualism*. Edward Arnold, 1987.

[3] H. Bodlaender and T. Kloks. A simple linear time algorithm for triangulating three-colored graphs. In *Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science*, pages 415–423, 1992.

[4] L.L. Cafalli-Sforza and M.W. Feldman. *Cultural transmission and evolution: A quantitative approach*. Princeton University Press, 1981.

[5] J. Camin and R. Sokal. A method for deducing branching sequences in phylogeny. *Evolution*, pages 311–326, 1965.

[6] L.L. Cavalli-Sforza. *Genes, Peoples, and Languages*. North Point Press, 2000.

[7] L.L. Cavalli-Sforza and A.W.F. Edwards. Phylogenetic analysis: Models and estimation procedures. *American Journal of Human Genetics*, pages 223–257, 1967.

[8] N. Chomsky. *Linguistics and cognitive science: Problems and mysteries*. 1991.

[9] N. Chomsky and M. Halle. *The Sound Pattern of English*. Harper and Row, 1968.

[10] M. Davenport and S.J. Hannahs. *Introducing Phonetics and Phonology*. Arnold, 1998.

[11] W.H.E. Day. Computationally difficult parsimony problems in phylogenetic systematics. *Journal of Theoretical Biology*, pages 429–438, 1983.

[12] M. Farach and S. Kannan. Efficient algorithms for inverting evolution. *Journal of the ACM*,

[13] M. Farach and M. Thorup. Fast comparison of evolutionary trees. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 481–488, 1994.

[14] J.S. Farris. Estimating phylogenetic trees from distance matrices. *American Naturalist*, pages 646–668, 1972.

[15] J.S. Farris. Distance data revisited. *Cladistics*, pages 67–85, 1985.

[16] J.S. Farris. Distances and statistics. *Cladistics*, pages 144–157, 1986.

[17] J. Felsenstein. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology*, pages 401–410, 1978.

[18] J. Felsenstein. Distance methods: a reply to farris. *Cladistics*, pages 130–144, 1986.

[19] J. Felsenstein. Phylogenies and molecular sequences: inference and reliability. *Annual Review of Genetics*, pages 521–565, 1988.

[20] J. Felsenstein. Phylip (phylogeny inference package). Published by author (Department of Genetics, University of Washington, Seattle), 2001.

[21] W.M. Fitch and E. Margoliash. The construction of evolutionary trees. *Science*, pages 29–94, 1976.

[22] T. Warnow F.R. McMorris and T. Wimer. Triangulating vertex colored graphs. In *Proceedings of the 4th Annual Symposium on Discrete Algorithms*, 1993.

[23] Jr. and F.R. McMorris G.F. Estabrook G.S. Johnson. A mathematical foundation for the analysis of character compatibility. *Mathematical Biosciences*, pages 181–187, 1976.

[24] H. Giegerich. *English Phonology: An Introduction*. Cambridge University Press, 1992.

[25] J. Goudsmit. *Viral Sex: the nature of AIDS*. Oxford University Press, 1997.

[26] R.L. Graham and L.R. Foulds. Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computational time. *Mathematical Biosciences*,

[27] D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, pages 19–28, 1991.

[28] M. Fellows H. Bodlaender and T. Warnow. Two strikes against perfect phylogeny. In *Proceedings of the 19th International Colloquium on Automata, LLanguages, and Programming*, Lecture Notes in Computer Science, pages 273–283. Springer Verlag, 1992.

[29] M. Haas. *The Prehistory of Languages*. Mouton, 1969.

[30] M.D. Hendy and D. Penny. Branch and bound algorithms to determine minimal evolutionary trees. *Mathematical Biosciences*, pages 277–290, 1982.

[31] M.D. Hendy and D. Penny. A framework for the quantitative study of evolutionary trees. *Systematic Zoology*, pages 297–311, 1989.

[32] H.M. Hoenigswald. *Linguistic Change and Language Reconstruction*. The University of Chicago Press, 1960.

[33] G. Hudson. *Essential Introductory Linguistics*. Blackwell Publishers, 2000.

[34] S. Kannan and T. Warnow. Triangulating three-colored graphs. *Siam Journal on Discrete Mathematics*, pages 249–258, 1992.

[35] S. Kannan and T. Warnow. A fast algorithm for the computation and enumeration of perfect phylogenies when the number of character states is fixed. *SIAM Journal on Computing*, pages 595–603, 1996.

[36] A. Kasher, editor. *The Chomskyan Turn*. Blackwell, 1991.

[37] T. Warnow M. Bonet C. Phillips and S. Yooseph. In *Proceedings of the Twenty-eighth annual ACM Symposium on the Theory of Computing*, pages 220–229, 1995.

[38] S. Kannan M. Farach and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, pages 155–179, 1993.

[39] L.E. Mettler and T.G. Gregg. *Population Genetics and Evolution*. Foundations of Modern

[40] J. Nichols. *Linguistic Diversity in Space and Time*. The University of Chicago Press, 1992.

[41] S. Pinker. *The Language Instinct: How the Mind Creates Language*. William Morrow and Company, 1994.

[42] O.W. Robinson. *Old English and Its Closest Relatives: A Survey of the Earliest Germanic Languages*. Stanford University Press, 1992.

[43] M. Ruhlen. *The Origin of Language: Tracing the Evolution of the Mother Tongue*. Wiley, 1994.

[44] N. Saitou and M. Nei. The neighbour-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, pages 406–425, 1987.

[45] M.A. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, pages 91–116, 1992.

[46] S.G. Thomason and T. Kaufman. *Language Contact, Creolization, and Genetic Linguistics*. University of California Press, 1988.

[47] J. Tischler. *Glottochronologie und Lexicostatistik*. Innsbrucker Beitrage zur Sprachwissenschaft, 1973.

[48] T. Warnow. Mathematical approaches to comparative linguistics. *Proceedings of the National Academy of Sciences*, pages 6485–6590, 1997.